

Online Dominating Set and Variations on Restricted Graph Classes

TECHNICAL REPORT no. 380, ETH Zürich, Dept. of Computer Science

Stephan Eidenbenz

Institute of Theoretical Computer Science, ETH Zürich, Switzerland
eidenben@inf.ethz.ch

Abstract We study online versions of MINIMUM DOMINATING SET, MINIMUM CONNECTED DOMINATING SET, and MINIMUM INDEPENDENT DOMINATING SET, where we restrict the input graphs to belong to a certain graph class after each insertion step. We show that straight-forward and easy-to-implement online strategies achieve optimum or nearly optimum competitive ratios for trees, unit disk graphs, and bounded degree graphs for standard and independent dominating sets. For connected dominating sets, our results are not tight and thus provide challenges for future research. In an orthogonal approach, we show that for many graph classes, such as planar, bipartite, and bounded treewidth graphs, the optimum achievable competitive ratios are $\Theta(n)$, where n is the number of graph vertices.

1 Introduction

MINIMUM DOMINATING SET is a classic graph problem that has been studied intensely in the literature. It consists of finding a minimum subset of the vertices of a given graph such that every vertex in the graph is either in the subset or has a neighbor in the subset to which it is connected by an edge. MINIMUM DOMINATING SET is known to be *NP*-hard. It can be approximated to within a logarithmic approximation ratio by a straight-forward greedy approximation algorithm and no polynomial time algorithm can achieve an approximation ratio that is better than this (upto low-order terms) [4].

In this paper, we study online versions of MINIMUM DOMINATING SET. The basic model is as follows: Graph vertices together with the edges to the previously inserted vertices are sequentially inserted by an adversarial player. Our task after each insertion is to select a dominating set with respect to the current graph. If we have decided at some point that a vertex v is in the dominating set, we cannot remove this vertex from the dominating set at a later stage. Our goal is to select vertices in such a way that the resulting dominating set is minimum. We call this model the insertion model.

In a more realistic model, we allow deletions as well as insertions: in each step a vertex is inserted or deleted by an adversarial player. Our task is again

to always select a dominating set without ever taking a vertex out of the dominating set except when it is removed by the adversary. We call this model the insertion/deletion model.

MINIMUM CONNECTED DOMINATING SET is a variation of MINIMUM DOMINATING SET, in which we need to find a minimum dominating set whose vertices induce a connected subgraph. Similar to the standard offline dominating set problem, MINIMUM CONNECTED DOMINATING SET is known to be approximable with a logarithmic approximation ratio, which is optimum [5]. The online version of connected dominating set on unit disk graphs (see below) is particularly interesting as it models backbone routing in mobile ad-hoc networks [1, 2]. In such a backbone routing model, some vertices are selected to belong to the backbone and every vertex in the network knows its closest vertex that sits on the backbone: the backbone forms a connected dominating set. Communication between any two vertices then takes place by first sending the message from the sender to its backbone neighbor, then sending the message along the backbone to the backbone neighbor of the receiver, and finally to the receiving vertex. The backbone should consist of as few vertices as possible in order to be effective. Our online version models the event that new vertices are added to the network or removed from the network.

MINIMUM INDEPENDENT DOMINATING SET is a variation, in which we need to find a minimum dominating set whose vertices form an independent set. The offline version of this problem is known to be inapproximable with an approximation ratio of $n^{1-\epsilon}$, where n is the number of graph vertices and $\epsilon > 0$ is an arbitrarily small constant.

In this paper, we present upper and lower bounds on the competitive ratio that any online algorithm can achieve. The competitive ratio of an online algorithm is defined as the worst-case ratio of the number of vertices in the dominating set that the online algorithm constructs vs. the minimum number of vertices in any dominating set that an offline algorithm could find if given the final graph. For an introduction to the general concept of online algorithms, see [3].

Online dominating set on general graphs has been studied before in the literature [6], where the authors show that no online algorithm can achieve a competitive ratio for dominating set that is strictly better than $n - 1$, where n is the number of graph vertices. MINIMUM CONNECTED DOMINATING SET and MINIMUM INDEPENDENT DOMINATING SET have not been studied in online versions before. We are interested in online dominating set variations on restricted graph classes, such as bounded degree graphs or trees. In our online model for a restricted graph class, we restrict the adversarial player in the sense that we require that the graph belongs to the restricted graph class after every insertion or deletion operation.

The first graph class that we study are trees: In Section 2, we first show that no online algorithm for dominating set on trees can guarantee to achieve a competitive ratio better than two by providing a counter-example. On the positive side, we propose a simple online algorithm that always finds solutions that

are optimum upto one vertex for connected dominating sets and that achieves a competitive ratio of three for standard dominating sets. This result is based on a proof that any minimum dominating set in a tree contains at least a third of all internal nodes of the tree. As a last result in this section, we show that no online algorithm can guarantee to beat a trivial competitive ratio of $n - 1$ for online independent dominating sets on trees.

The second graph class are unit disk graphs, which we study in Section 3. A unit disk graph is defined by unit size circles that are spread out in the plane. Each circle is represented by a vertex in the graph and two vertices are connected by an edge, if the two corresponding circles overlap or touch. We show that a straight-forward online algorithm achieves a competitive ratio of five, which is optimum as we also give a lower bound of five. These results even carry over to the insertion/deletion model and to the case, where we want to find independent dominating sets. Our result for connected dominating set are far from tight: an upper bound of $8 + \epsilon$ for an arbitrarily small $\epsilon > 0$ and a lower bound of $\frac{10}{3}$ is the best we can show.

B bounded degree graphs form the third graph class that we investigate. Parameter B indicates the maximum number of edges that can be connected to a graph vertex. In Section 4, we show that a simple online strategy achieves an optimum competitive ratio of B for dominating sets and independent dominating sets, which also extends to the insertion/deletion model. Again, our results for connected dominating sets are not tight, as we show a lower bound of $B - 1$ an upper bound of $B + 1$.

As a last set of graph classes, we consider classes, on which online dominating set seems to be as hard as on general graphs, i.e., the best ratios achievable are $n - 1$. In Section 5, we show that the following graph classes have this property: bounded tree width graphs, forests, bipartite graphs, planar graphs, and variable-size disk graphs. These strong bounds hold for standard and independent dominating sets. For connected dominating sets, we get a weaker lower bound of $n - 2$.

Most of our online algorithms are quite straight-forward and simple, but they achieve optimum, or nearly optimum competitive ratios. In fact, our online algorithms are very easy to implement and thus promise to achieve good results in different applications.

We will use the following notation in the insertion model: v_i is the i -th vertex inserted, G_i is the graph after the first i vertices have been inserted.

2 Trees

In this section, we first show that no online algorithm can guarantee to achieve a competitive ratio of two for online dominating set on trees. In our online model of dominating set on trees, vertices are inserted sequentially and the graph has to be a tree at each stage. Thus, when vertex v_i for $i > 1$ is inserted it is connected by an edge to exactly one vertex from G_{i-1} .

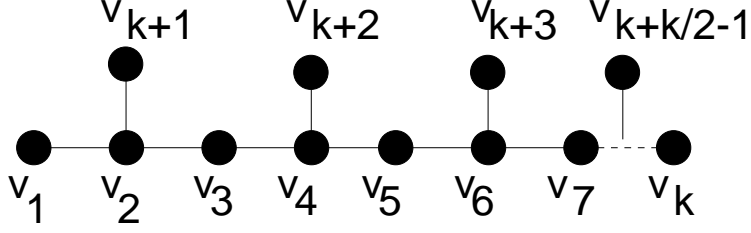


Figure 1. Lower bound for dominating set on trees

Lemma 1. *No online algorithm can guarantee to achieve a competitive ratio strictly better than two for online dominating set on trees.*

Proof. Consider the graph given in Fig. 1. The first k vertices v_1, \dots, v_k are inserted in a path. Any online algorithm will have to put at least $\frac{k}{2}$ of these vertices into the dominating set, otherwise the dominating set would not be feasible at some stage. Let C denote the set of vertices from v_1, \dots, v_k that the online algorithm has chosen not to put in the dominating set. Our adversarial player then adds leaf vertices to exactly the vertices in C . In Fig. 1, these vertices are for example v_2, v_4, v_6 . In total, at most $\frac{k}{2}$ leaf vertices are added, since at least $\frac{k}{2}$ vertices are already in the dominating set. The online algorithm will thus have to add $|C|$ additional vertices into the dominating set, yielding a final dominating set with k vertices. An optimum offline algorithm, on the other hand, would put all vertices from C into the dominating set and it would cover the vertices in between two vertices of C by selecting every third vertex among those vertices; if there is one or two vertices between two vertices from C , the optimum offline algorithm will not select any of them. Since at least one vertex is always between two vertices from C , the optimum offline algorithm will select at most $\frac{k}{2}$ vertices; thus no online algorithm can achieve a ratio better than two. \square

The following simple online algorithm will achieve a good competitive ratio for both connected dominating sets and standard dominating sets.

Definition 1. *The online greedy algorithm for trees checks upon insertion of vertex v_i for $i > 1$ whether v_i is already covered by the existing dominating set. If this is not the case, it adds the parent of v_i to the dominating set. When the first vertex v_1 is inserted, it is added to the dominating set.*

This algorithm puts all internal vertices of a resulting tree in the dominating set. The resulting dominating set contains no leaves from the tree except for v_1 , which might be a leaf. Thus, it is actually a connected dominating set. In terms of competitive ratio, we have the following for connected dominating sets:

Lemma 2. *The online greedy algorithm for trees produces a connected dominating set that contains at most one vertex more than the optimum connected dominating set.*

Proof. Let OPT be an optimum connected dominating set for a given tree. We may assume that OPT does not contain any leaves (except if the tree is either an isolated vertex or a single pair of vertices), since we could otherwise obtain a smaller connected dominating set by putting the parent of a leaf into the dominating set rather than the leaf itself. Since the graph is a tree, any pair of vertices from OPT is connected through a unique path, whose vertices have to be in the connected dominating set as well; thus, OPT must contain all non-leaf vertices. Our online algorithm, on the other hand, finds a connected dominating set that consists of all non-leaf vertices and possibly one leaf (i.e., the first vertex inserted). Thus, the lemma follows. \square

The online greedy algorithm also finds good standard dominating sets on trees:

Lemma 3. *The greedy online algorithm for trees achieves a competitive ratio of $3 - \epsilon$ for an arbitrarily small $\epsilon > 0$ for online dominating set.*

Proof. Let DS denote any dominating set on a given input tree. We show how to construct from DS a connected dominating set CDS with $|CDS| \leq 3|DS|$. To this end, we root the tree at an arbitrary node $r \in DS$ and we mark vertex r with color red. We then iteratively build a connected dominating set CDS , which will consist of all red vertices, as follows: let $C \subseteq DS$ denote the not yet colored vertices in DS , from which a path of length at most three exists to a red vertex. We then color all vertices from C and all vertices that lie on these short paths red. The procedure stops, when there are no more non-colored vertices left in DS . The red vertices form our connected dominating set CDS .

To see that this procedure always terminates, assume for the sake of contradiction that it does not terminate, i.e., at some point in the procedure, there are still non-colored vertices left in DS and none of them has a path of length at most three to a colored vertex. Let v_i be such a non-colored vertex from DS that has a path of minimum length (strictly larger than three) among all non-colored vertices from DS to any colored vertex, say v_j . Let v_l be the second but last non-colored vertex on the path from v_i to v_j . Obviously, v_l is neither covered by v_j nor by v_i , which implies that there must exist another vertex from DS that covers v_l . Let this vertex be v_t . Vertex v_t cannot be colored red, otherwise we would have two different paths from v_l to root r (i.e., one through vertex v_j and one through vertex v_t) and thus our graph would not be a tree. So, vertex v_t is not colored and it is a neighbor of v_l . Thus, the path from v_t through v_l to v_j is of length three and a contradiction to the assumption that no such path exists. Obviously, CDS is a feasible connected dominating set, and since we connected each vertex from DS to CDS by coloring at most three vertices, we have $|CDS| \leq 3|DS|$.

Since our online algorithm produces a connected dominating set that is optimum upto one unit (see Lemma 2), and since $|DS| \geq \frac{|CDS|}{3}$ also for the optimum connected and standard dominating set, the lemma follows. \square

The situation is quite different if we require the dominating set to be independent:

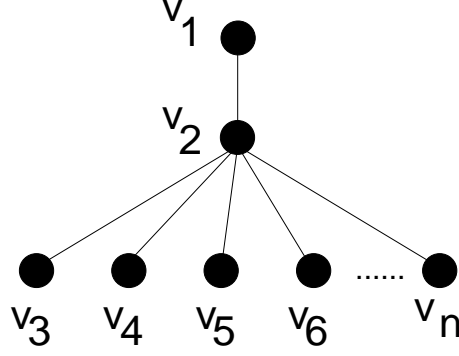


Figure2. Lower bound for independent dominating set on trees

Lemma 4. *No online algorithm can guarantee to achieve a competitive ratio of $n - 1$ for INDEPENDENT DOMINATING SET on trees.*

Proof. Consider the graph in Figure 2. Vertex v_1 must be placed into the independent dominating set by any online algorithm (otherwise the adversary stops after this step and we do not have a valid dominating set). Thus, vertex v_2 cannot be placed into the dominating set because it is connected to v_1 . Adding all the children of v_2 (i.e., vertices v_3, \dots, v_n) into the independent dominating set is the only feasible solution, yielding an independent dominating set with $n - 1$ vertices. An offline algorithm would simply put the single vertex v_2 into the dominating set. \square

A matching competitive ratio of $n - 1$ for independent dominating sets on trees is achieved by any online algorithm that produces a feasible solution. An example of such an optimum algorithm constructs an independent dominating set as follows: add a newly inserted vertex v_i to the independent dominating set if it is not yet covered by another vertex already in the solution.

3 Unit Disk Graphs

In this section, we study online versions of dominating set on unit disk graphs. We will use the terms vertex and disk interchangeably in this section. We first study the insertion model for the general dominating set and start by defining a straight-forward online greedy algorithm that will achieve a good competitive ratio.

Definition 2. *The online greedy algorithm for dominating set on unit disk graphs checks upon insertion of vertex v_i whether it is already covered by a vertex in the dominating set. If this is not the case, it adds v_i to the dominating set.*

This simple algorithm achieves an optimum competitive ratio:

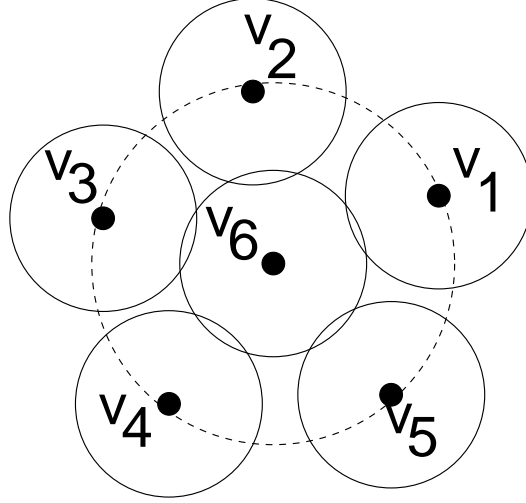


Figure3. Lower bound for unit disk graphs

Lemma 5. *The greedy algorithm for dominating set on unit disk graphs from Def. 2 achieves a competitive ratio of five.*

Proof. Let OPT be an optimum offline dominating set on the completely given graph G_n , let SOL be the dominating set computed by the online algorithm on G_n , let $v_i \in OPT$, and let $N(v_i)$ be the set of neighbor vertices of v_i in G_n . In the disk set $N(v_i) \cup v_i$, there can be at most five disks that do not touch each other, as it is well-known that unit disk graphs cannot have an induced $K_{1,6}$ [7], i.e., an induced subgraph that is bipartite and that has one vertex in one set that is connected to six independent vertices in the other set (also see Fig. 3 to get an intuitive understanding). Since the online algorithm never puts two disks into the dominating set that touch each other, it will put at most five disks from the disk set $N(v_i) \cup v_i$ into the dominating set. Note that for every vertex v_j from SOL (but not from OPT), there exists a vertex v_i from OPT that has v_j in its neighborhood, otherwise OPT would not be a feasible dominating set. Thus, for each disk in OPT , the solution SOL will contain at most five disks, yielding a competitive ratio of five. \square

Lemma 6. *No online algorithm or dominating set on unit disk graphs can achieve a competitive ratio strictly better than five.*

Proof. Consider Fig. 3. Disks v_1 through v_5 are inserted with their centers slightly inside a circle with radius two such that they do not touch each other. Disk v_6 is then inserted as a central disk that touches all five other disks. Any online algorithm will have to insert disks v_1 through v_5 into the dominating set, otherwise it would not have a valid dominating set after the first five insertions.

An optimum offline algorithm simply puts disk v_6 into the dominating set. We can create several copies of the basic gadget from Fig. 3, thus making this lower bound hold for any number of disks inserted. Therefore, no algorithm can guarantee a competitive ratio better than five. \square

If we allow deletions as well, we have a modified greedy algorithm:

Definition 3. *The online greedy algorithm for dominating set on unit disk graphs in the insertion/deletion model deals with insertions exactly as described in Def. 2. If a disk is deleted that was not in the dominating set, no action is taken. If a disk is deleted that was in the dominating set, we sequentially re-insert all disks that were neighbors of the deleted disk and that are not covered by other disks in the dominating set.*

In the insertion/deletion model, we require from an online algorithm with competitive ratio k that its solution at each step is not more than a factor k off the optimum offline solution of the current graph.

Lemma 7. *The online greedy algorithm for dominating set on unit disk graphs in the insertion/deletion model achieves a competitive ratio of five.*

Proof. Similar to the proof of Lemma 5, we can argue that the online algorithm guarantees at each stage that no two overlapping or touching disks are part of the dominating set. Thus, the solution of the online algorithm will always contain at most five disks for each disk in the optimum offline solution. \square

Since the lower bound from Lemma 6 obviously also holds in the insertion/deletion model, this algorithm is optimum.

Our greedy algorithm for dominating set on unit disk graphs from Def. 2 constructs an independent dominating set. The lower and the upper bound from Lemmas 5 and 6 as well as Lemma 7 for the insertion/deletion model extends to online independent dominating set on unit disk graphs in a straight-forward way. Thus:

Lemma 8. *The greedy algorithm for dominating set on unit disk graphs from Def. 2 finds an independent dominating set and achieves a competitive ratio of five for online independent dominating set, which is optimum. The online greedy algorithm for dominating set on unit disk graphs in the insertion/deletion model from Def. 3 finds an independent dominating set and achieves a competitive ratio of five for online independent dominating set.*

If we require our algorithms to find connected dominating sets, the situation is quite different.

Definition 4. *The online greedy algorithm for connected dominating set on unit disk graphs checks upon insertion of disk v_i whether it is already covered by a disk in the dominating set. If this is not the case, it adds a neighbor of v_i and v_i itself to the dominating set. Vertex v_1 is always in the dominating set.*

This algorithm yields a feasible dominating set as each neighbor of a newly inserted disk must have one of its neighbors in the dominating set. The authors of [1, 2] have developed a protocol for maintaining a connected dominating set in a unit disk graphs, when the vertices are moving. While their model is not exactly the same as ours (as we do not allow to take vertices out of a dominating set), their technique helps in the analysis of our algorithm.

Lemma 9. *The greedy algorithm for connected dominating set on unit disk graphs from Def. 2 achieves a competitive ratio of $8 + \epsilon$ for an arbitrarily small $\epsilon > 0$.*

Proof. Let MIS denote any maximal independent set in a unit disk graph and let OPT_{CDS} denote an optimum connected dominating set on the same unit disk graph. Then, according to [1], we have $|MIS| \leq 4|OPT_{CDS}| + 1$. We now look at our algorithm as selecting a maximal independent set and connecting the vertices of this set. Whenever our algorithm puts a vertex v_i into the dominating set immediately after it was inserted, we also add it to our independent set MIS ; the independent set MIS is actually an independent set, since we would not have added vertex v_i to it if it had already been covered by another vertex in the dominating set. For each vertex in MIS our algorithm adds one additional vertex (i.e., a neighbor of v_i) into the connected dominating set (except for v_1). Thus, the connected dominating set CDS that our algorithm finds is at most twice as large as the independent set MIS . The competitive ratio of our algorithm is thus:

$$\frac{|CDS|}{|OPT_{CDS}|} \leq \frac{2|MIS|}{\frac{|MIS|-1}{4}} = 8\left(1 + \frac{1}{|MIS|-1}\right),$$

which is $8 + \epsilon$ for an arbitrarily small $\epsilon > 0$. \square

The result from Lemma 9 matches the result from [1], where a distributed algorithm for computing a connected dominating set in a fixed graph was presented and an approximation ratio of 8 was shown.¹

The best lower bound for the connected case is quite far from the corresponding upper bound.

Lemma 10. *No online algorithm for connected dominating set on unit disk graphs can guarantee to achieve a competitive ratio strictly better than $\frac{10}{3}$.*

Proof. Consider Fig. 4. After insertion of vertex v_{10} , any online algorithm will have to have put at least vertices v_1, \dots, v_5 and v_7, \dots, v_9 into the dominating set. Inserting v_{11} will additionally force v_6 or v_{10} into the dominating set; inserting v_{12} will force v_{11} into the dominating set as well. Thus, for each of the k large circles consisting of eleven unit disks arranged around a central disk (which is inserted in a second phase only), any online algorithm will have to put at least

¹ Our result is in an online model, whereas the result from [1] is in a static setting; also, our proof seems to be simpler. However, the basic technique is similar.

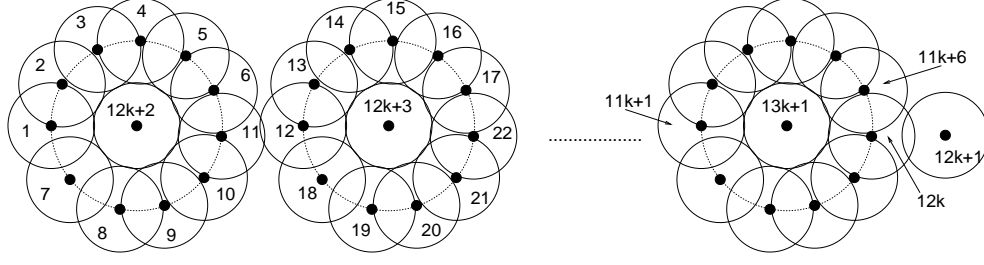


Figure 4. Lower bound for connected dominating set on unit disk graphs

10 vertices into the dominating set, yielding a connected dominating set of size at least $10k$. An optimum offline algorithm will simply put all central disks into the dominating set and connect them by adding two additional disks to the dominating set, thus yielding a connected dominating set of $3k - 1$ vertices. The lemma follows. \square

A close relative of the insertion/deletion model for connected dominating sets on unit disk graphs was studied previously [2], where a protocol was presented for maintaining a connected dominating set that is at most a factor 193 larger than the optimum connected dominating set.² Lemma 9 gives a lower bound for the best ratio achievable also in the model of [1].

4 Bounded Degree Graphs

Bounded degree graphs are a set of graph classes, where each class is characterized by a parameter $B > 1$, which indicates the maximum degree of a vertex (i.e., number edges connected to the vertex). A simple greedy strategy will yield the optimum result for these graph classes

Definition 5. *The online greedy algorithm for dominating set on B bounded degree graphs checks upon insertion of vertex v_i whether it is already covered by a vertex in the dominating set. If this is not the case, it adds v_i to the dominating set.*

Lemma 11. *The greedy algorithm for dominating set on B bounded degree graphs from Def. 5 achieves a competitive ratio of B .*

Proof. Let OPT be an optimum offline dominating set on the completely given graph G_n , let SOL be the dominating set computed by the online algorithm on G_n , let $v_i \in OPT$, and let $N(v_i)$ be the set of neighbor vertices of v_i in G_n . In the vertex set $N(v_i) \cup v_i$, there can be at most B vertices that the algorithm has added to the dominating set, since the algorithm never adds two neighboring

² There are considerable differences between the models, most prominently the model from [2] allows vertices to move in and out of the dominating set.

vertices to a dominating set and since v_i has at most B neighbors. Thus, for each vertex in OPT , the solution SOL will contain at most B vertices, resulting in a competitive ratio of B . \square

Lemma 12. *No online algorithm for dominating set on B bounded degree graphs can achieve a competitive ratio strictly less than B .*

Proof. Consider a graph G_B of B isolated vertices. Obviously, all B vertices must be in any valid dominating set. Vertex v_{B+1} is connected to vertices v_1, \dots, v_B . An optimum offline algorithm would of course only put vertex v_{B+1} into the dominating set. Since we can add as many copies of the graph G_{B+1} as we like, the lemma follows. \square

Along the lines of the previous section on unit disk graphs, a straight-forward extension of the online algorithm achieves the optimum competitive ratio of B also in the insertion/deletion model.

Since the greedy algorithm for dominating set on B bounded degree graphs from Def. 5 always produces independent dominating sets, Lemmas 11 and 12 as well as the extension to the insertion/deletion model also hold for independent dominating sets.

If we require connected dominating sets, the situation is again different: the upper and lower bounds are close, but not tight.

Definition 6. *The online greedy algorithm for connected dominating set on B bounded degree graphs checks upon insertion of vertex v_i whether it is already covered by a vertex in the dominating set. If this is not the case, it adds a neighbor of v_i to the dominating set. Vertex v_1 is always in the dominating set.*

This algorithm yields a feasible connected dominating set as each neighbor of a newly inserted vertex must have one of its neighbors in the dominating set.

Lemma 13. *The greedy algorithm for connected dominating set on B bounded degree graphs achieves a competitive ratio of $B + 1$.*

Proof. Let OPT be an optimum offline dominating set on the completely given graph G_n , let SOL be the dominating set computed by the online algorithm on G_n , let $v_i \in OPT$, and let $N(v_i)$ be the set of neighbor vertices of v_i in G_n . In the disk set $N(v_i) \cup v_i$, there can be at most $B + 1$ vertices that have been put into the connected dominating set, because the set only consists of at most this many vertices. The solution SOL cannot contain any vertices that are not neighbors of a vertex in OPT , otherwise, OPT would not be a dominating set. Hence, for each vertex in OPT , the solution SOL will contain at most $B + 1$ vertices, yielding a competitive ratio of $B + 1$. \square

Lemma 14. *No online algorithm for connected dominating set on B bounded degree graphs can achieve a competitive ratio strictly less than $B - 1$.*

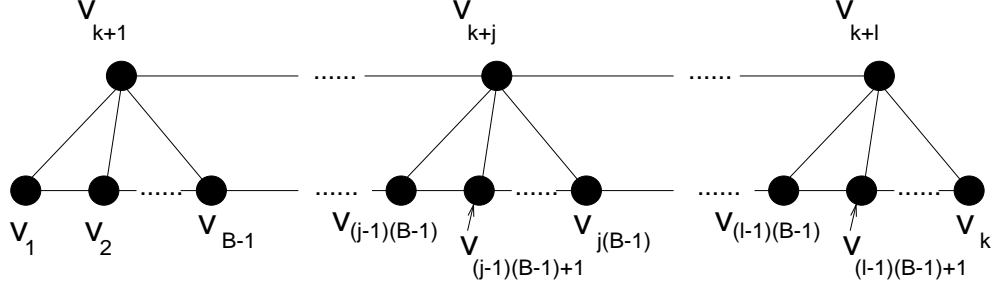


Figure 5. Lower bound for connected dominating sets on bounded degree graphs

Proof. Consider a graph G_k with $k = l \cdot (B-1)$ vertices for an integer l that lie on a path (i.e., v_i is connected to v_{i-1} when it is inserted). Any valid connected dominating set consists of $k-1$ vertices (i.e., all vertices except v_k). Vertices v_{k+j} for $j = 1, \dots, l$ are then inserted and connected to vertices $v_{(j-1)(B-1)}, \dots, v_{j(B-1)}$ and to vertex v_{k+j-1} . See Fig. 5. An optimum offline algorithm will only put vertices v_{k+j} for $j = 1, \dots, l$ into the connected dominating set. Thus, no online algorithm can achieve a competitive ratio of:

$$\frac{k-1}{l} = \frac{l(B-1)-1}{l} = B-1 - \frac{1}{l},$$

which is arbitrarily close to $B-1$ and thus the lemma follows. \square

5 Other Graph Classes

In this section, we discuss the online dominating set problem on other graph classes, where only a competitive ratio of $O(n)$ can be achieved by any online algorithm, where n is the number of inserted vertices.

Consider the graph in Fig. 6, in which first $n-1$ isolated vertices are inserted and then a last vertex is inserted that is connected to all other vertices. Any online algorithm will have to put the first $n-1$ vertices into the dominating set, whereas an offline algorithm will only put vertex v_n into the dominating set, hence resulting in a competitive ratio of $n-1$.

The graph in Fig. 6 is after each insertion stage a forest, has bounded tree width, is bipartite and planar. Thus, for all these graph classes, a simple online algorithm such as from Def. 5 that adds at most all but one vertex to the dominating set, is optimum. Moreover, the graph in Fig. 6 is also a variable-size disk graph, in which the first $n-1$ correspond to small disjoint equal-size disks and the last vertex corresponds to a large disk covering all other disks.

The example from Fig. 6 is also an independent dominating set, thus making the problem online independent dominating set as hard on these graph classes as the regular dominating set problem.

If we want connected dominating sets, consider Fig. 7, where $n-1$ vertices are inserted forming a path and an n -th vertex is inserted and connected to all

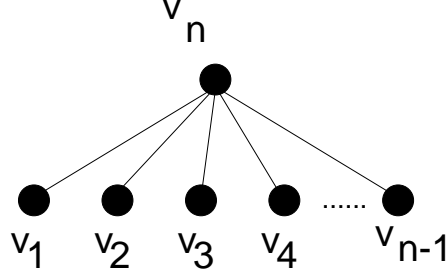


Figure6. Lower bound for dominating set on general graphs

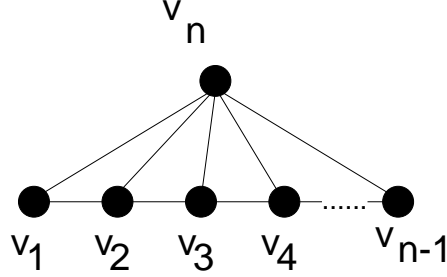


Figure7. Lower bound for connected dominating set on general graphs

other vertices. Any online algorithm will have to put $n - 2$ of the first $n - 1$ vertices into its connected dominating set, whereas an offline algorithm will only take vertex v_n . Thus, the best achievable competitive ratio is $n - 2$.

The graph in Fig. 7 is after each insertion stage a graph with bounded tree width, bipartite and planar. It is also the a variable-sized disk graph. Thus, for all these graph classes, a trivial online algorithm such as from Def. 6 that adds at most all but one vertex to the connected dominating set, is optimum upto one unit.

6 Conclusion

We have proposed numerous upper and lower bounds for online dominating set, connected dominating set and independent dominating set on several restricted graph classes. In most cases, it turns out that very simple online heuristics achieve optimum or nearly optimum competitive ratios. The following table, whose entries are competitive ratios and $\epsilon > 0$ is an arbitrarily small number, gives an overview of the results in this paper:

Graph class:	Dominating Set		Connected DS		Independent DS	
	lower	upper	lower	upper	lower	upper
Trees	2	$3 - \epsilon$	1	$1 + \epsilon$	$n - 1$	$n - 1$
Unit disk graphs	5	5	$\frac{10}{3}$	$8 - \epsilon$	5	5
Bounded degree graphs	B	B	$B - 1$	$B + 1$	B	B
Bounded treewidth, bipartite, planar, and variable-size disk graphs	$n - 1$	$n - 1$	$n - 2$	$n - 1$	$n - 1$	$n - 1$
Forests	$n - 1$	$n - 1$	n/a	n/a	$n - 1$	$n - 1$

Additionally, we showed that some of our results extend to the insertion/deletion model. This work leaves a few gaps for some graph classes that should be closed in future research. Moreover, it would be interesting to study online dominating set on other graph classes in order to get a complete picture as to where constant bounds can be achieved.

References

1. K. M. Alzoubi, P. Wan, O. Frieder; *Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks*; Proceedings IEEE INFOCOM 2002.
2. K. M. Alzoubi, P. Wan, O. Frieder; *Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks*; Proceedings MOBIHOC 2002.
3. Allan Borodin, Ran El-Yaniv; *Online Computation and Competitive Analysis*; Cambridge University Press, 1998.
4. P. Crescenzi, V. Kann; *A Compendium of NP Optimization Problems*; in the book by G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, *Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties*, Springer-Verlag, Berlin, 1999; also available in an online-version at: <http://www.nada.kth.se/theory/compendium/compendium.html>.
5. S. Guha and S. Khuller; *Approximation algorithms for connected dominating sets*; Algorithmica, 20:347–387, 1998.
6. Gow-Hsing King, Wen-Guey Tzeng; *On-line algorithms for the dominating set problem*; Information Processing Letters, Vol. 61, pp. 11 - 14, 1997.
7. M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi and D.J. Rosenkrantz; *Simple heuristics for unit-disk graphs*; Networks 25, pp. 59-68, 1995.